

TMVA Doubts

Specifying the Training Data:

For testing the optimizer, I had to create a DeepNet with only Dense Layers.
I have the following parameters,

```
size_t nSamples = 256;  
size_t nFeatures = 8;  
size_t nOutput = 1;  
size_t batchSize = 8;
```

```
size_t inputDepth = 1;  
size_t inputHeight = 1;  
size_t inputWidth = nFeatures;
```

```
size_t batchDepth = batchSize;  
size_t batchHeight = 1;  
size_t batchWidth = nFeatures;
```

```
DeepNet_t deepNet(batchSize, inputDepth, inputHeight, inputWidth, batchDepth, batchHeight, batchWidth,  
ELossFunction::kMeanSquaredError, EInitialization::kGauss, ERegularization::kNone, 0.0, true);
```

I have a doubt to construct the TensorDataLoader Object for the trainingData.

```
using TensorInput = std::tuple<const std::vector<TMatrixT<Double_t>> &, const TMatrixT<Double_t> &, const TMatrixT<Double_t> &>;  
using DataLoader_t = TTensorDataLoader<TensorInput, Architecture_t>;  
TensorInput trainingInput(XTrain, YTrain, WTrain);  
DataLoader_t trainingData(trainingInput, nSamples, batchSize, batchDepth, batchHeight, batchWidth, nOutput, nThreads);
```

So how should my XTrain, YTrain, WTrain dimensions be?

XTrain = `vector< TMatrixT<Double_t> >` = (1 x nSamples x nFeatures) or (nSamples x 1 x nFeatures)

YTrain = `TMatrixT<Double_t>` = (nSamples x nOutput)

WTrain = `TMatrixT<Double_t>` = (nSamples x 1)

And also I am not sure of these dimensions,

```
auto my_batch = trainingData.GetTensorBatch();  
my_batch.getInput() = vector<Matrix_t> = ( ? x ? x ? )  
my_batch.getOutput() = Matrix_t = ( ? x ? )  
my_batch.getWeights() = Matrix_t = ( ? x ? )
```